

FIG. 2

T06250" 66082460

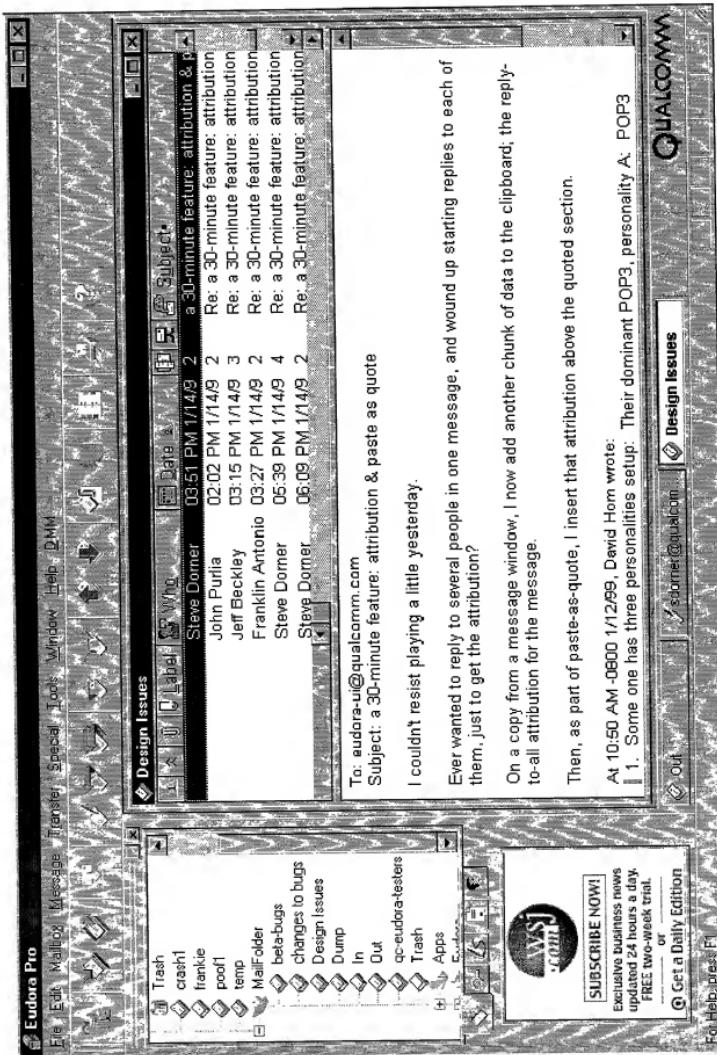


FIG. 3A

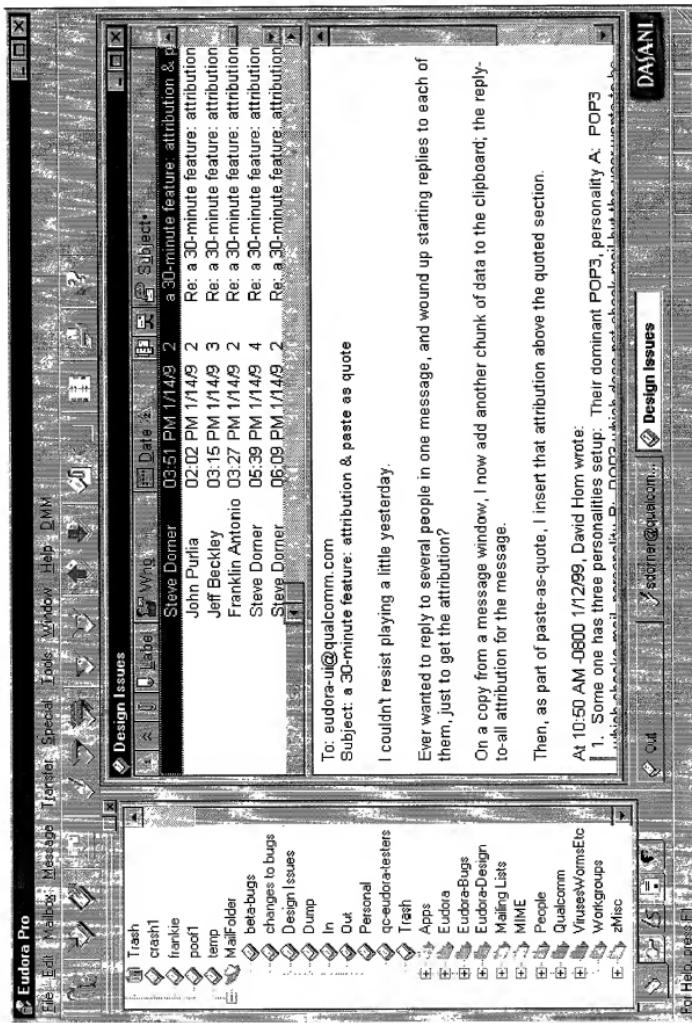


FIG. 3B

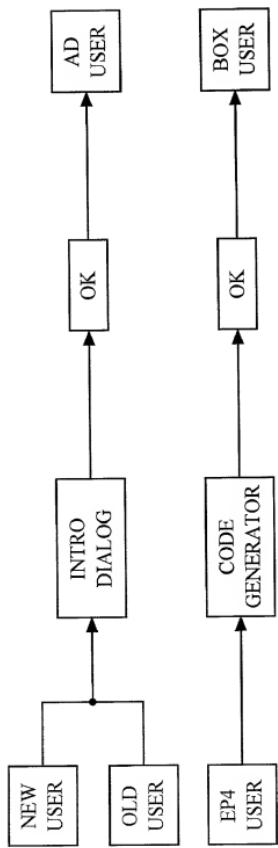


FIG. 4A

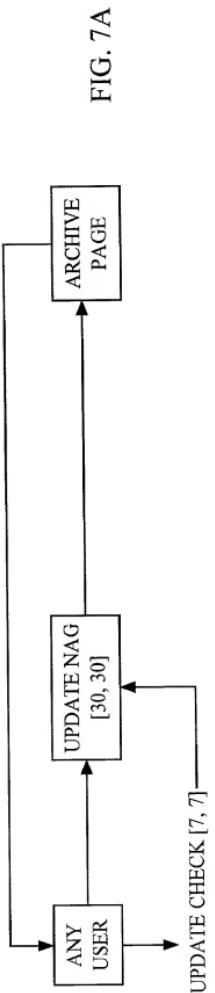


FIG. 7A

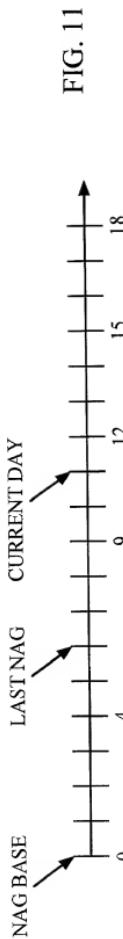


FIG. 11

WELCOME TO EUDORA!

EUDORA IS NOW LICENSED IN THREE WAYS; SPONSORED MODE, PAID MODE, AND LIGHT MODE. UNLESS YOU CHANGE MODES, EUDORA WILL RUN IN SPONSORED MODE, MEANING IT WILL DISPLAY ADS.

WE HAVE DONE OUR BEST TO PRESENT THE ADS IN A WAY THAT RESPECTS THE WORK YOU DO IN EMAIL. BY ALLOWING EUDORA TO DISPLAY ADS, YOU GET THE FULL POWER OF EUDORA FOR FREE AND WE CAN STILL PAY OUR BILLS.

IF YOU DECIDE THE ADS ARE NOT FOR YOU, YOU CAN CHANGE MODES. PAID MODE SHOWS NO ADS. CURRENT EUDORA PRO 4.X USERS WILL BE ABLE TO UPGRADE TO PAID MODE FOR FREE. OTHER USERS WILL BE ABLE TO PAY A LICENSE FEE TO GO TO PAID MODE. AT THIS STAGE IN TESTING, THE MACHINERY FOR PAID MODE IS NOT FULLY TESTED, AND PAID MODE IS UNAVAILABLE. LIGHT MODE ALSO SHOWS NO ADS, BUT HAS MANY FEWER FEATURES.

TO SWITCH FORMS OF EUDORA, PLEASE USE THE "PAYMENT & REGISTRATION" ITEM IN THE HELP MENU. TO LEARN MORE ABOUT THE THREE MODES, CLICK ON THE "TELL ME MORE" BUTTON BELOW.

TELL ME MORE

OK

FIG. 4B

706250 "66082460

7/42

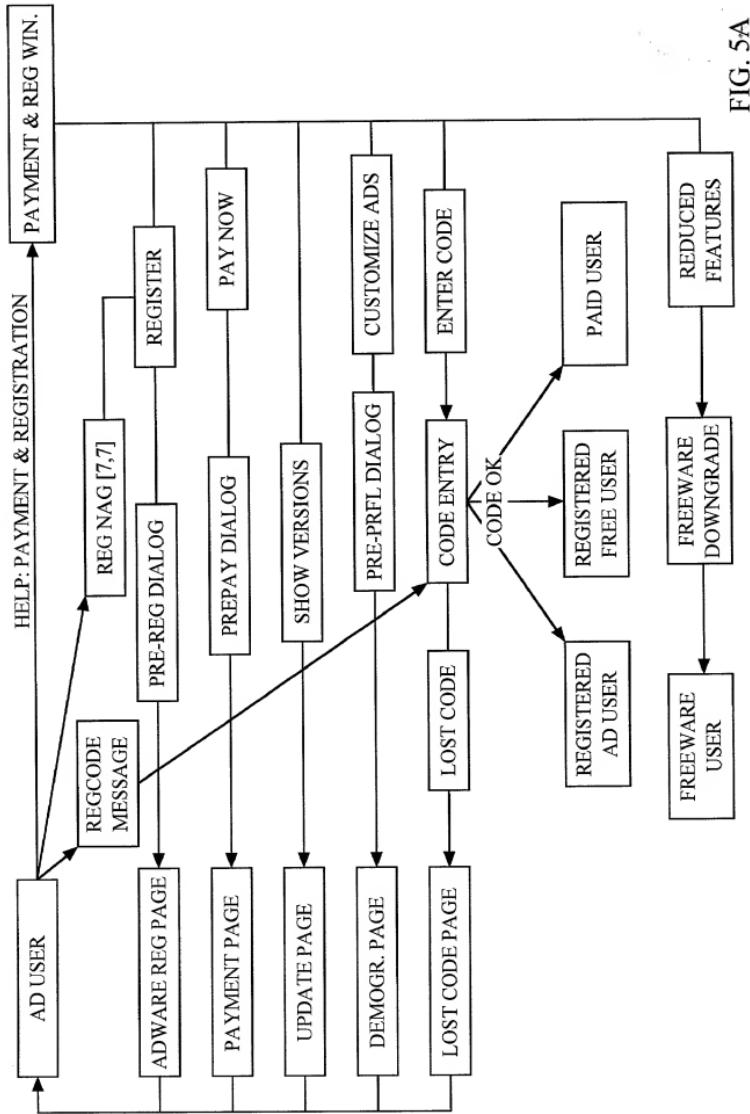


FIG. 5A

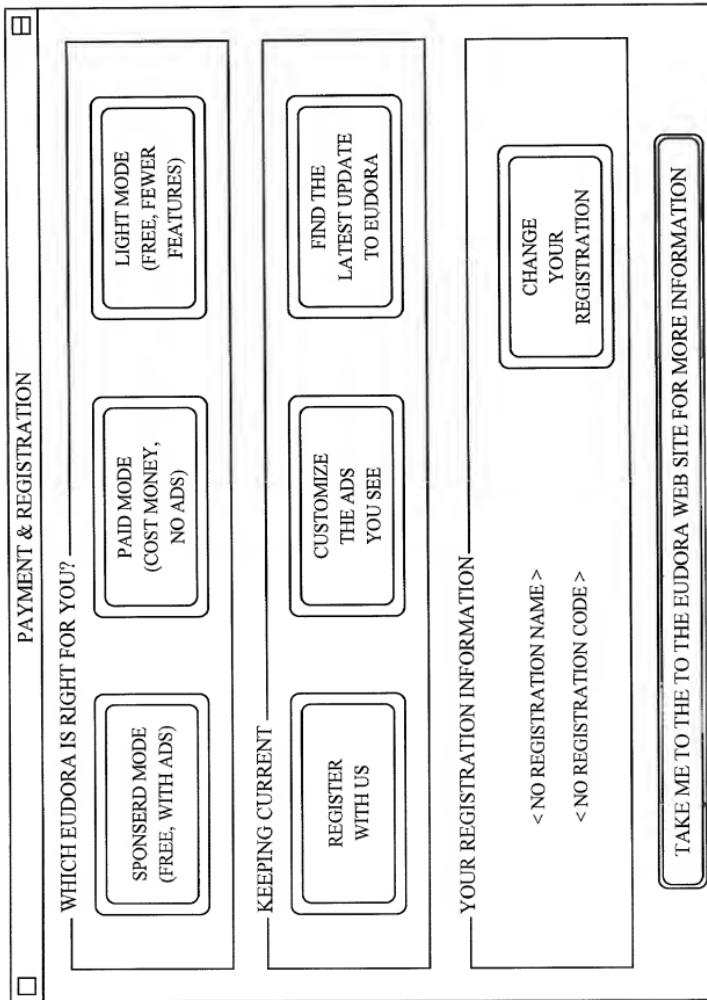


FIG. 5B

<p>WOULD YOU LIKE TO REGISTER YOUR COPY OF EUDORA ?</p> <p>AS A REGISTERED USER OF EUDORA WE WON'T NAG YOU AS OFTEN AS WE DO. WE'LL ALSO ERECT A GIANT STATUE IN YOUR IMAGE ON THE FRONT LAWN OF OUR CORPORATE HEADQUARTERS (*).</p> <p>HOW COOL IS THAT? C'MON... REGISTER! IT'S FUN AND EASY!</p> <p>(* GIANT STATUE OFFER VOID ON THE PLANET EARTH)</p>	<p>MAYBE LATER</p> <p>TAKE ME TO THE REGISTRATION PAGE</p>
--	--

FIG. 5C

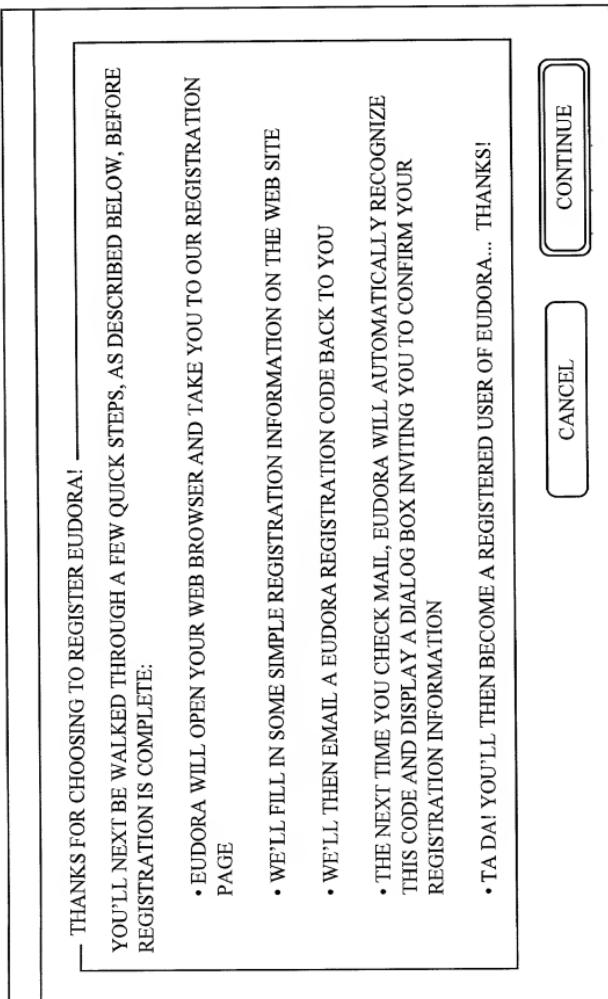


FIG. 5D

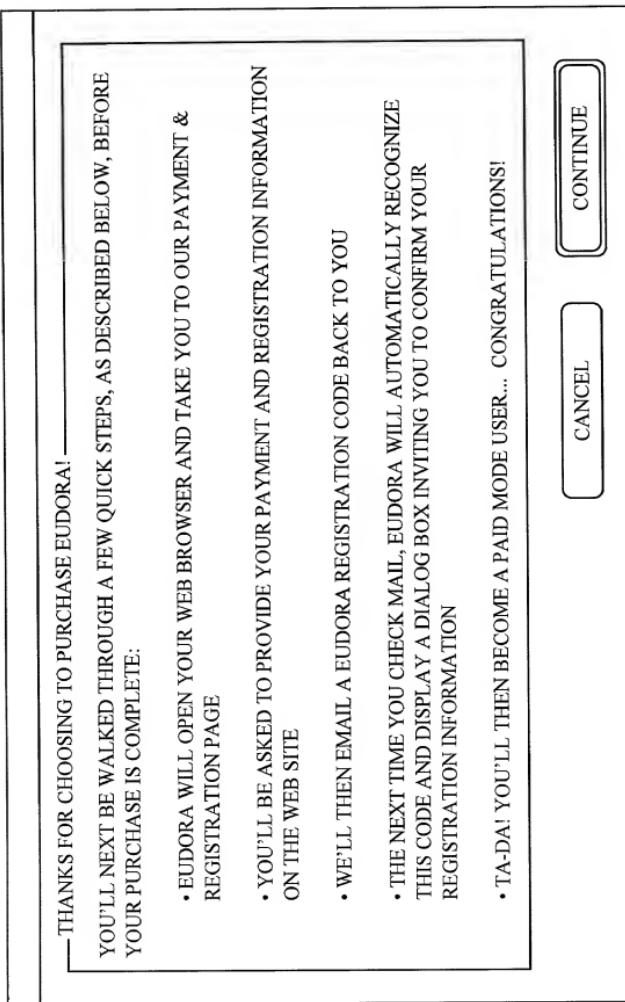


FIG. 5E

T06250 "65082760

12/42

THANK YOU FOR YOUR REGISTRATION!

TO COMPLETE YOUR REGISTRATION, PLEASE ENTER THE NAME YOU REGISTERED
UNDER AND YOUR REGISTRATION CODE BELOW.

THE EXACT NAME YOU REGISTERD UNDER:

FIRST NAME: JOHN

LAST NAME: MANTYARS

YOUR REGISTRATION CODE:

48925-89A2-B1149

I LOST THE CODE

FIG. 5F

□

□

□

DO YOU REALLY WANT TO SWITCH TO THE LIGHT VERSION OF EUDORA?

WHILE EUDORA IN LIGHT MODE REMAINS A VERY CAPABLE EMAIL CLIENT, IT LACKS THE POWER OF THE FULL VERSION. HERE ARE SOME OF THE FEATURES YOU WOULD BE GIVING UP, WITH CHECKMARKS NEXT TO THE ONES YOU'RE USING NOW:

◀ ▶

- ▢ b CHECK THE SPELLING OF YOUR EMAIL MESSAGES AS YOU TYPE
- ▢ b MULTIPLE PERSONALITIES FOR MANAGING MULTIPLE MAIL ACCOUNTS OR IDENTITIES
- ▢ b MESSAGE STATIONERY TO HELP YOU RESPOND TO YOUR MAIL MORE QUICKLY
- ▢ b MULTIPLE SIGNATURES TO HELP PERSONALIZE YOUR MAIL
- ▢ b MORE POWERFUL FILTERING

CHANGE THE PERSONALITY ASSOCIATED WITH MESSAGES FOR BETTER ORGANIZATION

- ▢ b PLAY VARIOUS SOUNDS WHEN MAIL ARRIVES DEPENDING ON YOUR FILTERS
- ▢ b OPEN A MESSAGE OR MAILBOX IN RESPONSE TO A FILTER
- ▢ b PRINT MAIL DIRECTLY FROM FILTERS

YOU CAN CONTINUE TO ENJOY THE TIME-SAVING POWER OF THESE FEATURES, AT NO CHARGE, SIMPLY BY LEAVING EUDORA IN SPONSORED MODE. IF YOU REALLY WANT TO MAKE THE ADS GO AWAY BUT KEEP EUDORA'S FULL CAPABILITIES, HIT "CANCEL" AND THEN SELECT "FULL VERSION (COSTS MONEY, NO ADS)."

▢ CANCEL

▢ REDUCED FEATURES

FIG. 5G

705250" 66082460

14/42

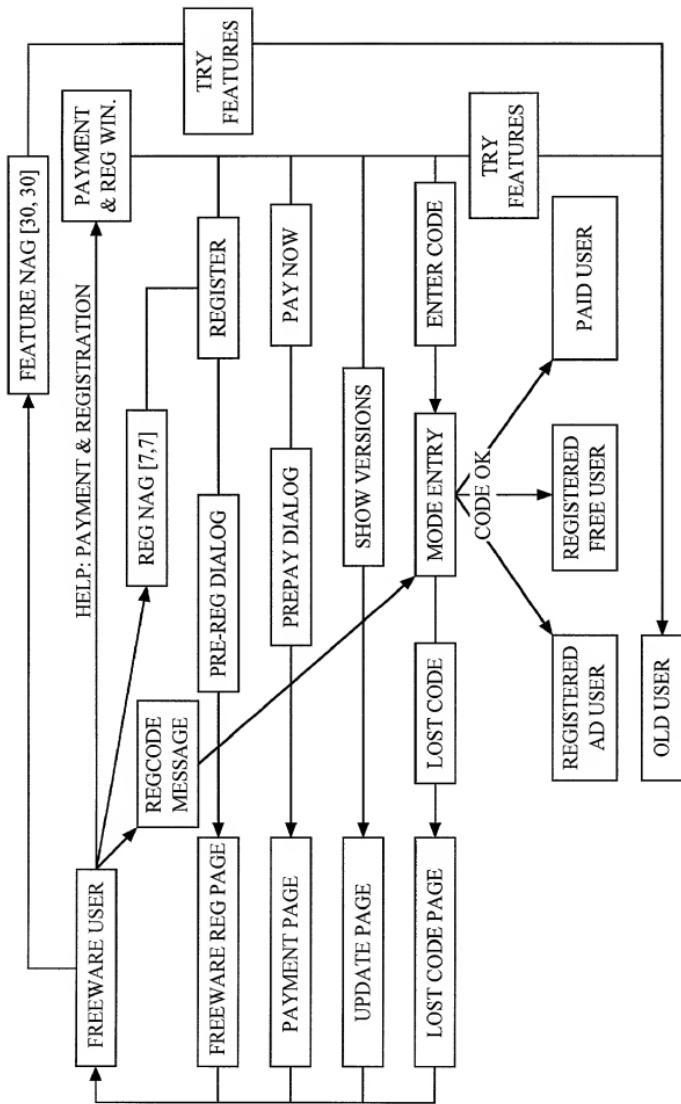


FIG. 6A

□ □ □

WOULD YOU LIKE TO TRY THE FULL-FEATURED VERSION OF EUDORA?

WHILE EUDORA IN LIGHT MODE REMAINS A VERY CAPABLE EMAIL PROGRAM, IT LACKS ALL THE POWER OF THE FULL VERSION. HERE ARE SOME OF THE CAPABILITIES YOU COULD BE USING TO MANAGE YOUR EMAIL (AND YOU'LL BE GETTING MORE OF IT, WE'RE SURE). THE FULL VERSION IS FREE BECAUSE IT IS SPONSOR-SUPPORTED. THAT MEANS IT HAS ADS IN IT, BUT THEY ARE DISPLAYED IN A WAY THAT'S SENSITIVE TO WHAT YOU'RE DOING WHEN YOU'RE IN EMAIL.

► [] []

CHECK THE SPELLING OF YOUR EMAIL MESSAGES AS YOU TYPE
MULTIPLE PERSONALITIES FOR MANAGING MULTIPLE MAIL ACCOUNTS OR IDENTITIES
MESSAGE STATIONERY TO HELP YOU RESPOND TO YOUR MAIL MORE QUICKLY
MULTIPLE SIGNATURES TO HELP PERSONALIZE YOUR MAIL
MORE POWERFUL FILTERING
CHANGE THE PERSONALITY ASSOCIATED WITH MESSAGES FOR BETTER ORGANIZATION
PLAY VARIOUS SOUNDS WHEN MAIL ARRIVES DEPENDING ON YOUR FILTERS
OPEN A MESSAGE OR MAIL BOX IN RESPONSE TO A FILTER
PRINT MAIL DIRECTLY FROM FILTERS

► []

THESE FEATURES WILL BE TURNED ON AUTOMATICALLY, AT NO CHARGE, WHEN YOU CLICK ON THAT ENTICING BUTTON BELOW. (C'MON...TAKE A CHANCE.)

► []

WOW! I WANT TO TRY ALL THE FEATURES!

CANCEL

FIG. 6B

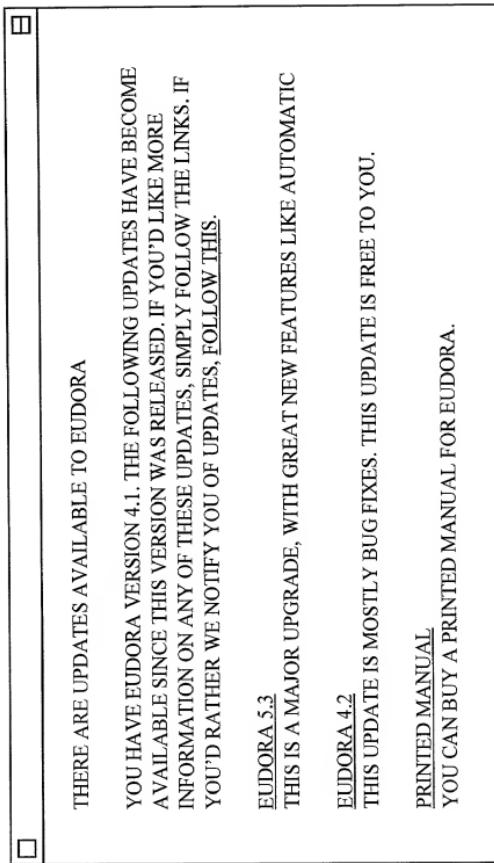


FIG. 7B

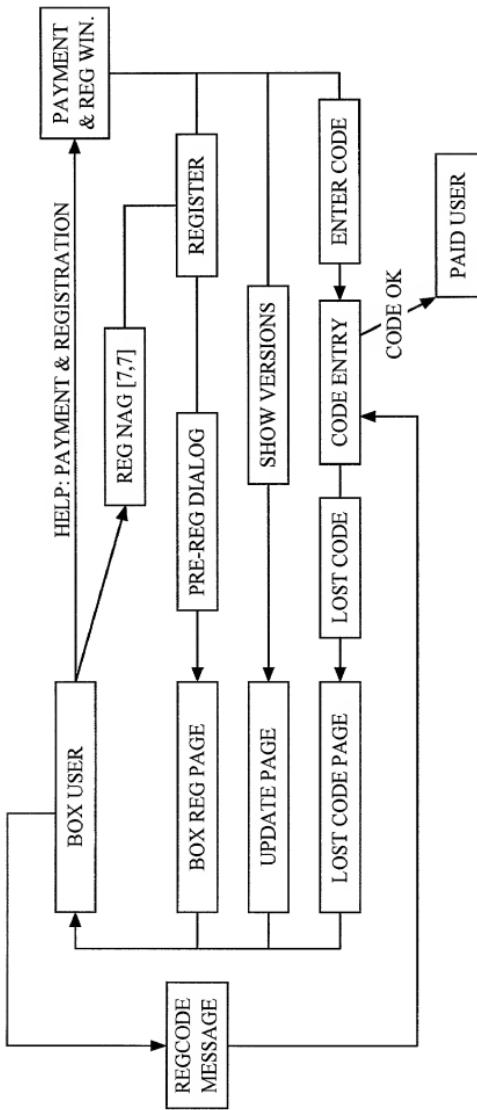


FIG. 8

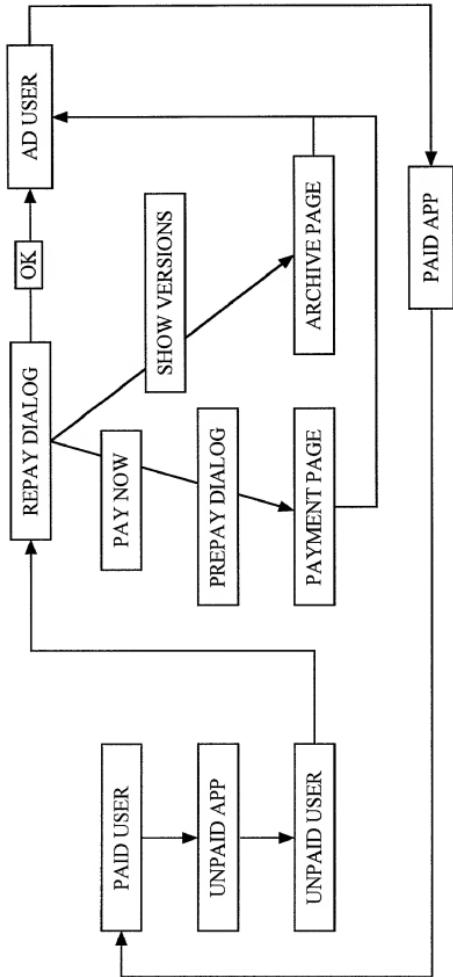


FIG. 9

705250-66082460

19/42

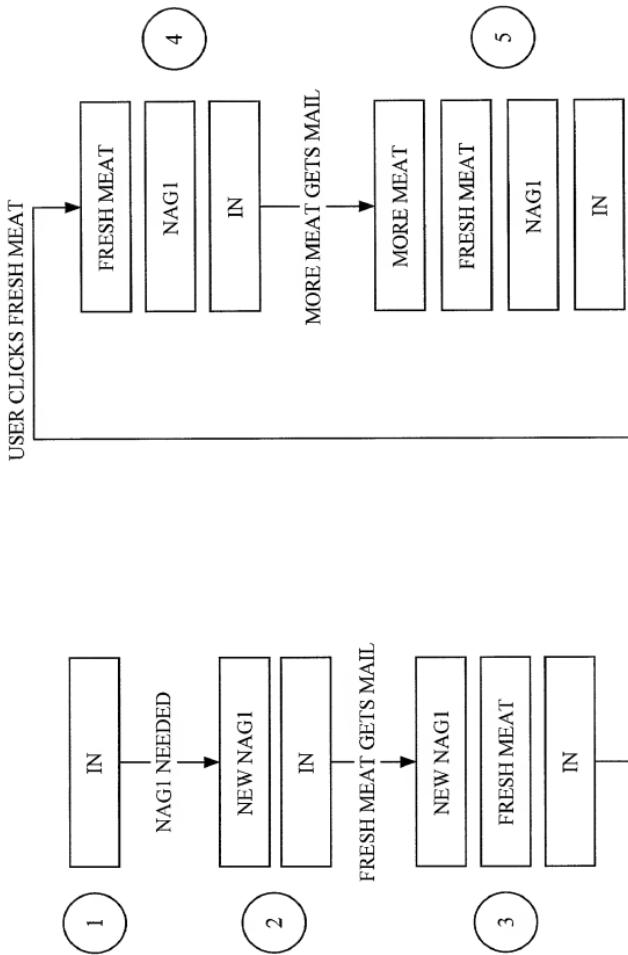


FIG. 10

706250" 660824610

20/42

LINK HISTORY		
TYPE	SITE	DATE VISITED
□	APPLE COMPUTER	WED, SEP. 1, 1999, 4:48 PM
□	FTP.QUALCOMM.COM/EUDORA	TODAY, 11:26 AM
□	QUALCOMM STORE	WED, SEP. 1, 1999, 4:48 PM
□	MAC OS RUMORS	ASAP!
□	MDUDZLAK@QUALCOMM.COM	TODAY, 11:23 AM
□	WWW.QUALCOMMONES.PRODUC...	WED, SEP. 1, 1999, 4:48 PM
□	WWW.EUDORA.COM	ATTEMPTED

◀ ▶

VIEW REMOVE

FIG. 12A

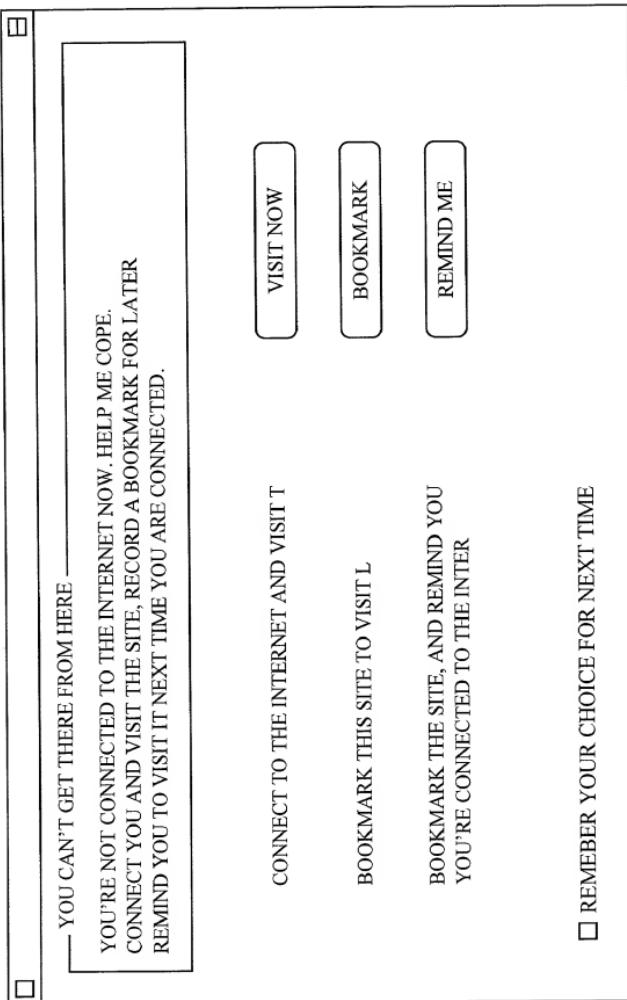


FIG. 12B

ASSUMPTIONS	
AVERAGE CONNEC. SPEED, Xbps	28.8
AVERAGE AD SIZE, Xbps/cm	9.3
NUMBER OF USERS	8,000,000
NUMBER OF HOURS RUNNING EUDORA	2
NUMBER MAILCHECKS PER USER PER HOUR	2
PLAYLIST ENTRY SIZE, BYTES	500

FIG. 13A

IMPLICATIONS						
# OF NEW ADS PER USER PER DAY	# SECONDS DOWN-LOADING ADS	# SECONDS AD BAND-WIDTH, Xbps/AD CHECK	8x USERS AD BAND-WIDTH, Xbps	AD Xbps/100,000 USERS	Avg. CONN-EXCTIONS, 1000'	8 x USERS PLAYLIST BAND-WIDTH, Xbps
10	26	6	67	0.8	2.4	4
15	39	10	101	1.3	3.6	5
20	52	13	135	1.7	4.8	7
25	65	16	168	2.1	6.0	9
30	78	19	202	2.5	7.2	11
35	90	23	235	2.9	8.4	12

FIG. 13B

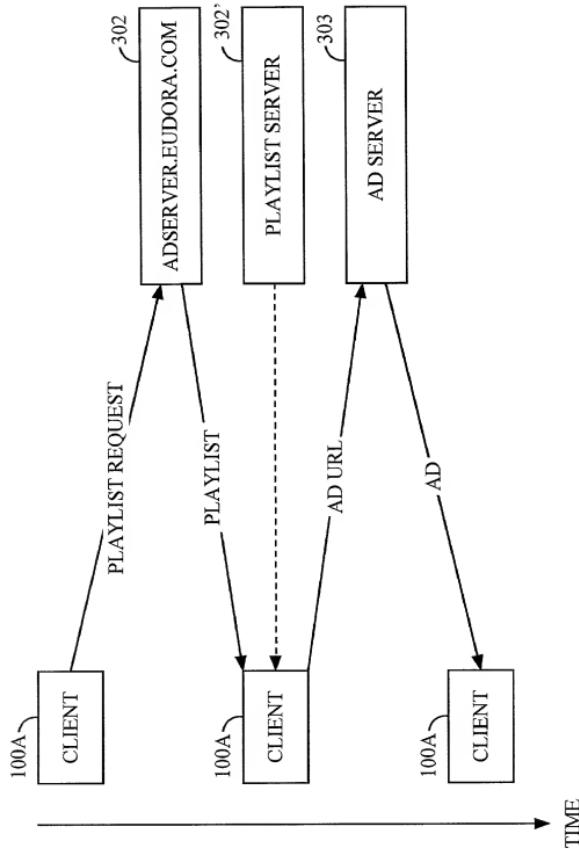


FIG. 14

```
//////////  
// Main ad scheduler  
ScheduleMain  
{  
// Has a new day dawned?  
Do CheckForNewDay  
// Are we within the current ad's showFor?  
if ( ad.thisShowTime < ad.showFor )  
{  
// there is nothing to be done  
return  
}  
// At this point, we know that we need a new ad  
// Perform housekeeping tasks on the old one  
do AdEndBookkeeping  
// Pop out of a block if all ads on par  
if ( block isn't all playlists )  
{  
find ad with minimum ad.numberShown  
if ( ad.numberShown >= blockGoal )  
set block to all playlists  
}  
// If we are over our quota of regular ads for the day,  
// look for a runout  
if ( adFaceTimeToday > faceTimeQuota )  
{  
do ShowARunout  
}  
else  
{  
Do ShowARegularAd  
}  
}  
}  
// end ad schedule main
```

```
||||||||||||||||||||||||||

// We must perform certain tasks when the calander day
changes.
CheckForNewDay
{if ( the calander day has changed )
{
// Perform housekeeping tasks on the ad currently showing
Do StopShowingCurrentAd
// Runout ads are charged for a full showFor if they've been
shown
// at all on a given day. Charge any runout ads if they've
been
// shown at all.
for runout ads
{
if ( ad.thisShowTime > 0 )
{
ad.totalTimeShown += ad.showFor
ad.thisShowTime = 0
}
}
// Now, reset the counters for all ads to reflect the fact
that
// a new day has dawned.
for all ads
{
ad.numberShownToday = 0
}
// Record yesterday's facetime
// Might not literally be yesterday, be sure to use
// whatever day the app was last run on
set old current day's facetime to totalFaceTimeToday
// and reset our global regular ad facetime counter
adFaceTimeToday = 0
totalFaceTimeToday = 0
// if we were in a block, back out
set block to all playlists
}
}
// end CheckForNewDay
```

||||||||||||||||||||||||||

```
// This function shows a runout ad, and if it
// can't find one, goes to a rerun
ShowARunout
{
    for runout ads
    {
        // has the ad been flushed?
        if ( ad.flushed )
            try next ad
        // are we done showing this runout today?
        if ( ad.numberShownToday > ad.dayMax )
            try next ad // this one's used up for the day
        // are we done showing this runout for ever and ever?
        if ( ad.shownFor > ad.shownForMax )
            try next runout ad // this one's used up forever
        // are we between the ad's start and end date?
        if ( ad.startDate < the current date < ad.endDate )
            try next runout ad
        // the ad is not supposed to run today
        // do we actually HAVE the ad?
        if ( ad has not been downloaded )
        {
            ask for ad to be downloaded
            try next ad
        }
        // ok, we believe we should show this runout
        // we are now in runout state
        Do ShowAnAd
        return
    }
    // if we haven't found a runout ad, we will go to "rerun"
    state
    Do ShowARerun
}
// end ShowARunout
```

FIG. 15C

||||||||||||||||||||||||||||

```
// Rerun state. Look for a regular ad to rerun
ShowARerun
{
    for regular ads [ in current block ]
    {
        // has the ad been flushed?
        if( ad.flushed )
            try next ad
        // is this ad recent enough to rerun?
        if( ad.lastShownDate is older than returnInterval )
            try next ad
        // this one is too old to rerun
        // if in block, show ads only if it's their "turn"
        if( ad.numberShownToday >= blockGoal )
            try next ad // need to find a friend in this block
        // are we between the ad's start and end dates?
        if( ad.startDate < the current date < ad.endDate )
            try next ad
        // the ad is not supposed to run today
        // do we actually HAVE the ad?
        if( ad has not been downloaded )
        {
            ask for ad to be downloaded
            try next ad
        }
        // ok, at this point we can show this ad, but because
        // we're in rerun, we don't keep the books
        Do ShowAnAd
        return
    }
    // if we get here, we have no ads to show. Punt.
    return
}
// end ShowARerun
```

FIG. 15D

```
///////////
// Show a regular ad
ShowARegularAd
{
for regular ads [ in current block ]
{
// has the ad been flushed?
if ( ad.flushed)
try next ad
// are we done showing this ad today?
if (ad.numberShownToday > ad.dayMax )
try next ad // this one's used up for the day
// if in block, show ads only if it's their "turn"
if ( ad.numberShownToday >= blockGoal )
try next ad // need to find a friend in this block
// are we done showing this ad for ever and ever?
if ( ad.shownFor > ad.showForMax )
try next ad // this one's used up forever
// are we between the ad's start and end dates?
if ( ad.startDate < the current date < ad.endDate )
try next ad
// the ad is not supposed to run today
// do we actually HAVE the ad?
if ( ad has not been downloaded )
{
ask for ad to be downloaded
try next ad
}
// ok, we believe we should show this ad
// we are now in regular state
Do ShowAnAd
return
}
// if we get here, we have failed to find a regular
// ad. Go to runout
Do ShowARunout
}
// end ShowARegularAd
```

FIG. 15E

```
///////////
// Perform necessary housekeeping when we're taking
// down an ad
AdEndBookkeeping
{
// In rerun state, we don't do any bookkeeping
if ( inRerunState )
return
// Account for at most ad.showFor seconds, provided
// we've shown the ad for at least ad.showFor seconds
// Note that this means we don't charge for time beyond
// ad.showFor seconds, which is important
if ( ad.thisShowTime >= ad.showFor )
{
ad.numberShownToday += ad.showFor
ad.shownFor++
// we do NOT reset thisShowTime here, we do it in
// AdStartBookkeeping. It actually doesn't matter where
// we do it, provided we are careful NOT to do it for
// runout ads.
}
}
// end AdEndBookkeeping
```

FIG. 15F

```
//////////  
// Show an ad, including bookkeeping and block handling  
ShowAnAd  
{  
// If the ad is in a block, notice that  
if ( it's in a "block" playlist )  
{  
if ( not currently in a block )  
{  
find ad in block with minimum numberShown  
make that our ad  
set blockGoal to minimum numberShown+1  
}  
set current block to this playlist  
}  
// now do bookkeeping  
Do AdStartBookkeeping  
// and actually show it  
Do DisplayThatAd  
}
```

FIG. 15G

```

// Perform housekeeping when we put up an ad
AdStartBookkeeping
{
// In return state, we don't do any bookkeeping
if ( inRerunState )
return
// For regular ads
if ( it's a regular ad )
{
ad.thisShowTime = 0
ad.lastShownDate = now
}
}
// end AdStartBookkeeping

```

FIG. 15H

PERSISTANT ADS			
PLAYLIST REQUEST	FACE TIME	USED TO DETERMINE HOW MUCH ADVERTISING	
	FACE TIME LEFT	TO SEND TO CLIENT	
		NOT USED	
PLAYLIST RESPONSE CLIENTINFO	REQINTERVAL	RELATIVELY LARGE; ONE OR MORE DAYS	
	FLUSH	USED. SINGLE PLAYLIST COMPLETELY SPECIFIES	
		LIST OF ADS CLIENT SHOULD HAVE	
PLAYLIST RESPONSE SCHEDULING PARAMETERS	SHOWFORMAX		
		NOT USED	

FIG. 16A

SHORT-LIVED ADS			
PLAYLIST REQUEST	FACE TIME	NOT USED	
	FACE TIME LEFT	USED TO DETERMINE HOW MANY ADS CLIENT	
		SHOULD RECEIVE	
PLAYLIST RESPONSE CLIENTINFO	REQINTERVAL	NOT USED. INSTEAD, CLIENT REQUEST NEW	
	FLUSH	PLAYLIST WHENEVER ADS "RUN LOW".	
PLAYLIST RESPONSE SCHEDULING PARAMETERS	SHOWFORMAX	NOT USED	
		USED TO DETERMINE HOW LONG AN AD RUNS	

FIG. 16B

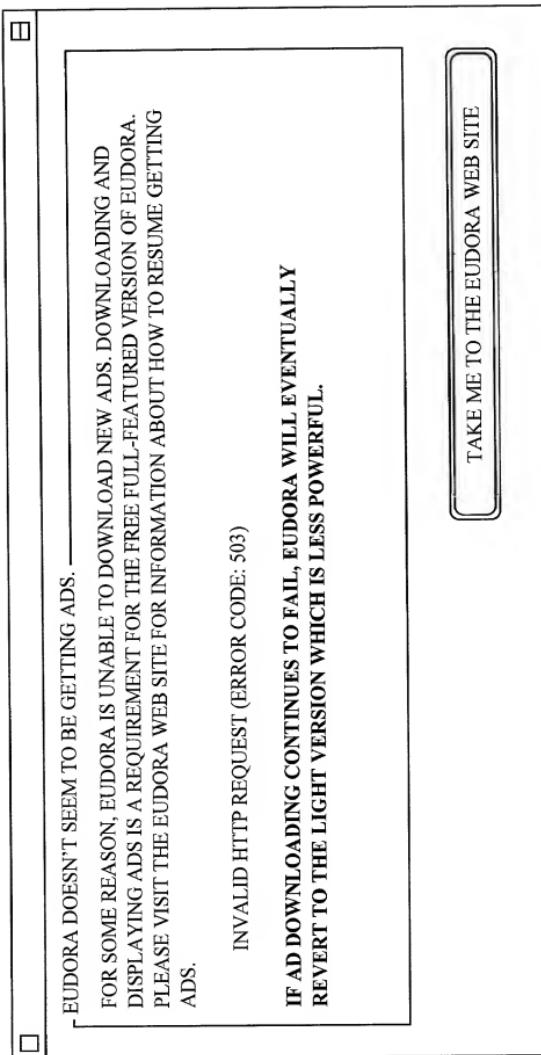


FIG. 17A

SOMETHING SEEKS TO BE COVERING THE AD.

IT'S PROBABLY INADVERTENT, BUT EUDORA HAS DETERMINED THAT YOU ARE COVERING UP ALL OR A SIGNIFICANT PORTION OF AN AD. THE SOFTWARE IS DESIGNED TO NOTIFY YOU WHEN THIS HAPPENS IN THE HOPES THAT YOU WILL STOP COVERING UP THE AD. IF YOU DON'T, THIS WINDOW WILL KEEP POPPING UP (WHICH YOU WILL PROBABLY FIND QUITE ANNOYING).

WE'VE ALWAYS GOT SOME GOOD STUFF UNDER DEVELOPMENT BACK AT THE HOME OFFICE, AND IT'S THE ADVERTISING IN EUDORA THAT ENABLES US TO CONTINUE TO DEVELOP THE SOFTWARE WHILE PROVIDING IT TO YOU FOR FREE. WE'VE WORKED HARD TO MAKE SURE THE ADVERTISING ISN'T ANNOYING, AND WE GENUINELY HOPE THAT YOU ARE NOT DELIBERATELY TRYING TO COVER THE ADS BECAUSE THEY'RE BOthering YOU. OF COURSE, YOU CAN CHOOSE TO PAY US FOR EUDORA BY CHOOSING "PAYMENT & REGISTRATION" FROM THE "HELP" MENU AND CLICKING ON "PAID FULL VERSION." OR YOU CAN REMOVE WHATEVER IS OBSCURING THE AD.



FIG. 17B

— EUDORA WILL NOW REVERT TO A LESS POWERFUL VERSION. —

EUDORA HAS BEEN UNABLE TO DOWNLOAD ADS FOR QUITE SOME TIME AND WILL NOW REVERT TO A LESS POWERFUL VERSION. IF YOU WOULD LIKE MORE INFORMATION ABOUT WHY EUDORA'S FEATURES ARE BEING REDUCED AT THIS TIME, PLEASE VISIT THE EUDORA WEB SITE. YOU WILL FIND INFORMATION THERE ABOUT HOW THE FULL-FEATURED VERSION CAN BE REACTIVATED.

WE'RE SORRY FOR THIS INCONVENIENCE.

[TAKE ME TO THE EUDORA WEB SITE](#)

SADLY, OK...

FIG. 17C

706250 * 66082460

<p>WE'D LIKE TO KNOW HOW YOU USE EUDORA.</p> <p>IN ORDER TO MAKE EUDORA WORK AS WELL AS POSSIBLE, IT'S IMPORTANT THAT WE KNOW HOW PEOPLE USE IT. WE ASK USERS FOR THIS INFORMATION AT RANDOM. LOOKS LIKE IT'S YOUR TURN. IF YOU'RE OPEN TO HELPING US THIS WAY, ALL YOU HAVE TO DO IS CLICK "GENERATE INFO" BELOW AND A MESSAGE WILL BE CREATED. YOU CAN REVIEW THE CONTENTS OF THE MESSAGE IF YOU LIKE, AND THEN SEND IT TO US OR NOT -- THAT'S UP TO YOU.</p> <p>WE VALUE OUR PRIVACY; WE'RE PRETTY SURE YOU VALUE YOURS, SO WE WANT YOU TO KNOW WHAT WE'LL BE COLLECTING AND GIVE YOU A CHANCE TO ELIMINATE ANYTHING YOU DON'T WANT TO SEND. SIMPLY UNCHECK THE BOXES NEXT TO ANY INFORMATION YOU'D RATHER NOT SEND.</p> <p>PLEASE UNDERSTAND THAT AS SOON AS WE RECEIVE YOUR EMAIL, WE WILL THROW AWAY THE HEADERS THAT IDENTIFY THE MAIL AS COMING FROM YOU. YOU SEE, WE DON'T ACTUALLY NEED TO KNOW WHO YOU ARE TO FIND YOUR INFORMATION HELPFUL. SO WE PROMISE TO PROTECT YOUR PRIVACY AND TURN YOU INTO "JUST A NUMBER." :-)</p>	<p>IT'S OK TO TRANSMIT STATISTICS REGARDING:</p> <p><input type="checkbox"/> YOUR NET/EUDORA USAGE <input type="checkbox"/> YOUR DEMOGRAPHIC DATA <input type="checkbox"/> ADVERTISEMENT INFORMATION <input type="checkbox"/> NON-PERSONAL SETTINGS</p> <p><input type="button" value="CANCEL"/> <input type="button" value="GENERATE INFO"/></p>
--	---

FIG. 18A

FIG. 19

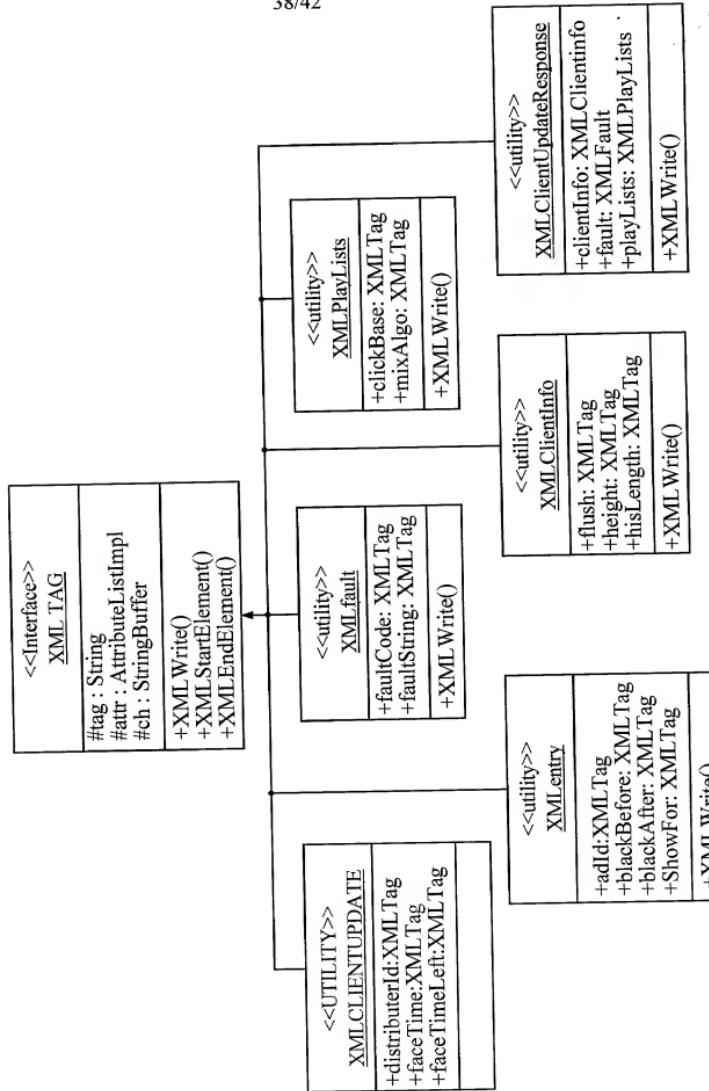


FIG. 20

```
% The list of available ads advantageously can be built from the following query:  
  
ads = dbCon.prepareStatement("SELECT" * FROM ads WHERE StartDate<=today AND endDate>=today + 30 AND  
AdType = "I" AND AdStatus = "A" AND ImpressionsServed<Impressions ORDERED BY ImpressionsServed ASC);  
  
run out ads = dbCon.prepareStatement("SELECT" * FROM ads WHERE StartDate <= today AND endDate >= today +  
30 AND AdType = "R" AND AdStatus = "A" AND ImpressionsServed<Impressions ORDERED BY ImpressionsServed  
ASC);
```

% The time required to deliver the ads advantageously can be calculated in the following manner.

face time left for today [seconds] = faceTime[today] - faceTimeUsedToday

(Comment: Face time left for today is the number of seconds the servlet can use to
deliver special ads today.)

predict face time [seconds] = SUM(faceTime[tomorrow], faceTime[tomorrow+1], ... faceTime[tomorrow+reqInterval])
goal show time left [seconds] = predict face time -faceTimeLeft

(Comment: Predict face time is the number of seconds the servlet predicts the user is
going to have.)

(Comment: Goal show time left is the number of seconds that the software provider needs to fill
with ads.)

FIG. 21A

```

% Targeting
while (face time left for today) {
    if ad is not in the history {
        select ad [according to target = today]
        face time left for today -= ad.showFor
    }
    next ad
}

while (Goal show time left ) {
    if ad is not in the history {
        select ad [according to target]
        goal show time left -= ad.showFor
    }
    next ad
}

```

Default values:

reqInterval = 1 day.
 facetime = 30 minutes
 faceTimeQuota is ?
 histLength = 31 days

FIG. 21B

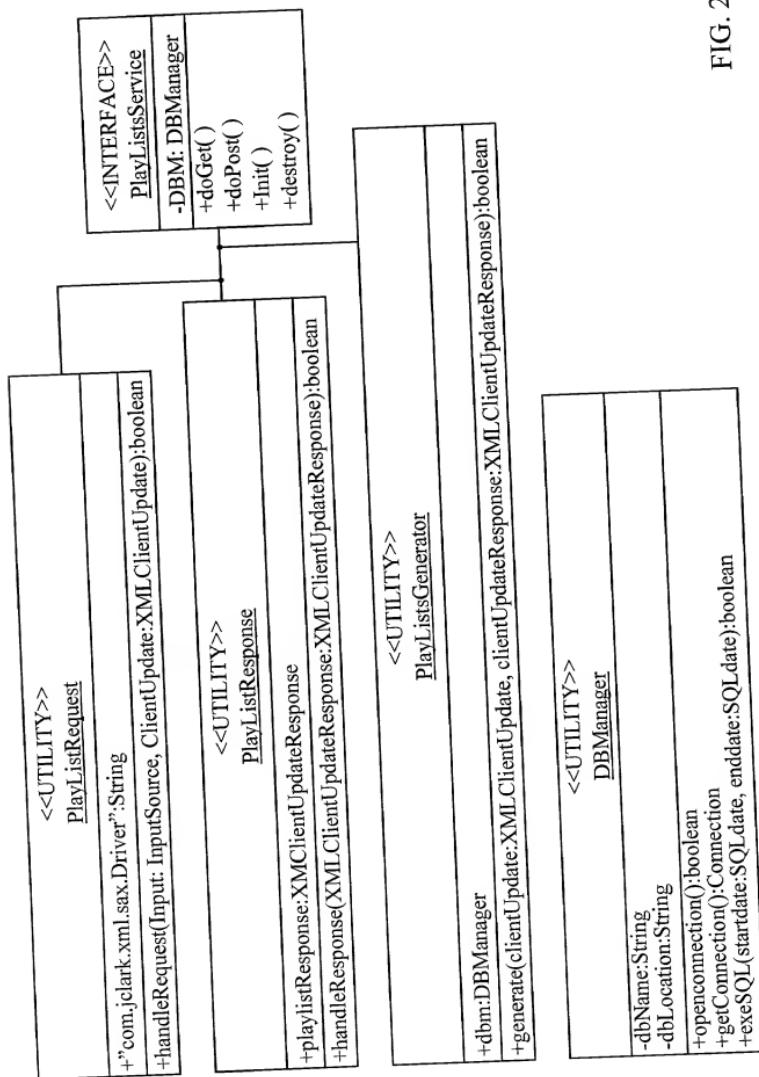


FIG. 22

